**PROJECT DATABASE API FUNCTIONS**

**db\_open()**

Open connection to the database

**db\_close()**

Close connection to the database

**get\_snp\_by\_id(query\_string)**

query\_string snp\_id to query

return dataframe of results (empty if no results)

SELECT snp\_id, chromosome, position, p\_value, mapped\_gene, phenotype, population

FROM SNP\_Associations

WHERE snp\_id LIKE "%<query\_string>%""

ORDER BY snp\_id

LIMIT 20

**get\_snp\_by\_gene(query\_string)**

query\_string mapped\_gene to query

return dataframe of results (empty if no results)

SELECT snp\_id, chromosome, position, p\_value, mapped\_gene, phenotype, population

FROM SNP\_Associations

WHERE mapped\_gene LIKE "%<query\_string>%""

ORDER BY snp\_id

LIMIT 20

**get\_snp\_by\_coordinates(chromosome, start, end)**

chromosome snp\_chromosomeid to query

start starting chromosome position to query from

end ending chromosome position to query to

return dataframe of results (empty if no results)

SELECT snp\_id, chromosome, position, p\_value, mapped\_gene, phenotype, population

FROM SNP\_Associations

WHERE chromosome = "<chromosome>" AND position >= <start> AND position <= <end>

ORDER BY snp\_id

LIMIT 20

**get\_gene\_annotations\_by\_gene\_symbol(query\_string)**

query\_string gene\_symbol to query

return dataframe of results (empty if no results)

SELECT gene\_symbol, gene\_id, chromosomal\_locus, snp\_id, pathway, go\_term, category, specificity

FROM Gene\_Annotations

WHERE gene\_symbol LIKE "%<query\_string>%""

ORDER BY gene\_symbol

LIMIT 20

**get\_gene\_annotations\_by\_gene\_id(query\_string)**

query\_string gene\_id to query

return dataframe of results (empty if no results)

SELECT gene\_symbol, gene\_id, chromosomal\_locus, snp\_id, pathway, go\_term, category, specificity

FROM Gene\_Annotations

WHERE gene\_id LIKE "%<query\_string>%""

ORDER BY gene\_id

LIMIT 20

**get\_gene\_annotations\_by\_snp(query\_string)**

query\_string snp\_id to query

return dataframe of results (empty if no results)

SELECT gene\_symbol, gene\_id, chromosomal\_locus, snp\_id, pathway, go\_term, category, specificity

FROM Gene\_Annotations

WHERE snp\_id LIKE "%<query\_string>%""

ORDER BY snp\_id

LIMIT 20

**def get\_population\_by\_name(query\_string)**

query\_string population\_name to query

return dataframe of results (empty if no results)

SELECT sample\_name, sex, biosample\_id, population\_code, population\_name, superpopulation\_code, superpopulation\_name, population\_elastic\_id, data\_collections

FROM Population

WHERE population\_name LIKE "%<query\_string>%""

ORDER BY sample\_name

LIMIT 20

**def get\_snp\_and\_gene\_by\_snp(query\_string)**

query\_string snp\_id to query

return dataframe of results (empty if no results)

SELECT SNP\_Associations.snp\_id as snp\_associations\_snp\_id, chromosome, position, p\_value, mapped\_gene, phenotype, population, gene\_symbol, gene\_id, chromosomal\_locus, Gene\_Annotations.snp\_id as gene\_annotations\_snp\_id, pathway, go\_term, category, specificity

FROM SNP\_Associations LEFT JOIN Gene\_Annotations ON SNP\_Associations.snp\_id = Gene\_Annotations.snp\_id

WHERE SNP\_Associations.snp\_id LIKE "%<query\_string>%""

ORDER BY SNP\_Associations.snp\_id

LIMIT 20

**def get\_snp\_and\_population\_by\_snp(query\_string)**

query\_ string snp\_id to query

return dataframe of results (empty if no results)

SELECT snp\_id, chromosome, position, p\_value, mapped\_gene, phenotype, population, sample\_name, sex, biosample\_id, population\_code, population\_name, superpopulation\_code, superpopulation\_name, opulation\_elastic\_id, data\_collections

FROM SNP\_Associations LEFT JOIN SNP\_Associations.population = Population.population\_name

WHERE snp\_id LIKE "%<query\_string>%""

ORDER BY snp\_id

LIMIT 20

**def** **get\_summary\_stats\_by\_population(query\_string)**

query\_ string population to query

return dataframe of results (empty if no results)

SELECT population, tajimas\_d, xp\_ehh, his, nucleotide\_diversity

FROM Selection\_Stats

WHERE population LIKE "%<query\_string>%""

ORDER BY population

LIMIT 20

**def** **get\_summary\_stats\_by\_snp(query\_string)**

query\_ string: snp\_id to query

return dataframe of results (empty if no results)

SELECT population, tajimas\_d, xp\_ehh, his, nucleotide\_diversity

FROM SNP\_Associations INNER JOIN Selection\_Stats ON SNP\_Associations.population = Selection\_Stats.population

WHERE snp\_id LIKE "%<query\_string>%""

ORDER BY snp\_id

LIMIT 20